

Pengukuran Kinerja Sistem *Publish/Subscribe* Menggunakan Protokol MQTT (*Message Queuing Telemetry Transport*)

Salsalina Oktaria F. Tarigan^{#1}, Herry Imanta Sitepu^{#2}, Maclaurin Hutagalung^{#3}

[#]*Departemen Sistem Komputer, Institut Teknologi Harapan Bangsa
Jln. Dipatiukur no. 80-84, Bandung*

¹*lily_tarigan@yahoo.co.id*

²*hisitepu@gmail.com*

³*maclaurin@gmail.com*

Abstrak— Salah satu model sistem terdistribusi yang digunakan untuk komunikasi antar perangkat adalah dengan sistem *publish/subscribe* (PUB/SUB). Ada beberapa protokol yang dapat digunakan untuk melakukan komunikasi *publish/subscribe*, salah satunya adalah protokol MQTT. Protokol MQTT adalah protokol yang didesain dengan karakteristik sederhana, ringan, dan mudah untuk diimplementasikan. Dengan demikian MQTT dapat diimplementasikan di perangkat dengan sumber daya komputasi yang terbatas. MQTT memiliki 3 level QoS, yaitu QoS 0, 1, dan 2. Penelitian ini mengukur dampak dari QoS protokol MQTT terhadap QoS jaringan, yaitu *bandwidth*, *delay*, *jitter*, dan *packet loss*. Hasil dari penelitian tersebut menunjukkan QoS pada protokol MQTT berpengaruh pada QoS jaringan, yaitu *bandwidth*, *delay*, *jitter*, dan *packet loss* pada jaringan *subscriber*.

Kata Kunci— *Internet of Things, Publish/Subscribe, Sistem Terdistribusi, Protokol MQTT, Quality of Service*

Abstract— *One of the distributed system model used for communication between devices is a publish/subscribe. There are some protocols that can be used for communication in publish/subscribe. MQTT protocol is one of them. MQTT protocol is a messaging protocol that is designed with characteristics such as simple, lightweight, and easy to implement. Thus, MQTT can be implemented in the devices with limited computing resources. MQTT has 3 levels of QoS, i.e. QoS 0, 1, and 2. This research measured the impact of MQTT protocol QoS in network QoS, i.e. bandwidth, delay, jitter, and packet loss. The result shows that MQTT protocol QoS impacts of network QoS, i.e. bandwidth, delay, jitter, and packet loss on the network subscriber.*

Keywords— *Internet of Things, Publish/Subscribe, Distributed System, MQTT Protocol, Quality of Service*

I. PENDAHULUAN

Menurut [3], tahun 2020 diprediksikan jumlah perangkat yang terhubung ke internet adalah sebanyak 50 milyar perangkat. Perangkat yang terhubung ke internet tersebut tidak hanya *smartphone* atau *tablets* tetapi benda-benda yang sering digunakan manusia dalam kehidupan sehari-hari, misalnya mobil. Mobil dapat mengetahui kapan harus mengisi bahan bakar, terjadi suatu kecelakaan pada rute yang digunakan, dan menyalakan mesin mobil lebih awal untuk mencairkan es pada mesin sisa badai es. Contoh-contoh ini merupakan salah satu

bentuk dari *Internet of Things*. IBM menyatakan bahwa “*Internet of Things* merupakan miliaran perangkat pintar yang saling berhubungan, bergerak, dan terkadang bertindak secara sendirinya [10].”

Salah satu model komunikasi yang digunakan dalam sistem *Internet of Things* adalah *publish/subscribe* (PUB/SUB). PUB/SUB merupakan salah satu model sistem terdistribusi yang terdiri dari *publisher* dan *subscriber*. *Publisher* adalah pengirim pesan sementara *subscriber* adalah penerima pesan berdasarkan langganan (*subscription*). *Broker* berfungsi sebagai perantara *publisher* dan *subscriber* untuk melanjutkan pesan dari *publisher* ke *subscriber* dan mengelola *subscription* dari *subscriber*.

Terdapat beberapa protokol yang digunakan untuk komunikasi sistem *publish/subscribe*, dua contoh protokol tersebut adalah MQTT dan DDS. Protokol MQTT adalah protokol yang awalnya digunakan untuk *telemetry*, sistem data yang terpusat, dan *connection oriented* sedangkan protokol DDS adalah protokol yang bersifat *intelligent system*, sistem data yang tersebar, dan *connectionless*. Protokol MQTT memiliki 3 level QoS, yaitu 0, 1, dan 2. Penelitian ini akan menggunakan protokol MQTT dalam pengukuran kinerja (*Quality of Service*) komunikasi *telemetry*.

Delay, *jitter*, *packet loss*, dan *bandwidth* merupakan parameter QoS jaringan yang dapat menyebabkan kondisi jaringan menjadi tidak ideal dan hal ini dapat mengganggu komunikasi antar perangkat. Karakteristik jaringan LAN memiliki *delay*, *jitter*, dan *packet loss* yang kecil serta *bandwidth* yang besar. Penelitian ini mengukur dampak dari penggunaan QoS protokol MQTT dalam kondisi jaringan yang memiliki *delay*, *jitter*, *packet loss*, dan *bandwidth*. Hasil akhir dari pengukuran QoS pada sistem *publish/subscribe* berupa grafik.

II. KAJIAN PUSTAKA

Teori-teori terkait tentang penelitian ini adalah sebagai berikut:

A. *Internet of Things*

Internet of Things adalah sebuah revolusi teknologi komputasi dan komunikasi di masa yang akan datang yang didasarkan pada konsep kapan dan dimana saja dapat saling terhubung. IBM dalam sebuah jurnal yang berjudul “*The Interconnecting of Everything*” menyatakan “*Internet of Things* merupakan internet masa depan, menggerakkan, dan memproses jutaan perangkat industri dan global [4].”

B. Sistem Terdistribusi

Sistem terdistribusi adalah sebuah sistem yang *software* atau *hardware*-nya terletak di dalam jaringan komputer dan saling berkomunikasi menggunakan *message passing*. *Message passing* adalah teknik untuk melakukan sinkronisasi dan komunikasi antara proses-proses yang saling berinteraksi [6]. Sistem terdistribusi terdiri dari dua jenis komunikasi, yaitu komunikasi langsung dan komunikasi tidak langsung. Komunikasi langsung adalah komunikasi tanpa perantara, sedangkan komunikasi tidak langsung adalah komunikasi menggunakan perantara.

C. Indirect Communication

Komunikasi tidak langsung yang terjadi pada sistem terdistribusi disebut juga dengan *indirect communication*. *Indirect communication* adalah komunikasi yang terjadi antara suatu *entity* dalam sebuah sistem terdistribusi melalui sebuah perantara tanpa alat penghubung antara pengirim dan penerima. *Indirect communication* banyak digunakan dalam komunikasi sistem terdistribusi, misalnya: *group communication*, *publish/subscribe system*, *message queue system*, *shared memory approaches*. Pada penelitian ini sistem yang terdistribusi tidak langsung yang digunakan adalah *publish/subscribe*.

D. Sistem Publish/Subscribe

Sebelum sistem *publish/subscribe* digunakan, komunikasi yang dilakukan adalah *point-to-point*, seperti terlihat pada Gambar 1.

Kompleksnya sistem komunikasi saat ini membuat sistem komunikasi *point-to-point* menjadi kurang efektif. Sistem *publish/subscribe* menjadi salah satu alternatif solusi. Sistem *publish/subscribe* adalah sistem yang terdiri dari *publishers* dan *subscriber*. *Publishers* akan menerbitkan suatu *event* dan *subscriber* menyampaikan ketertarikan kepada *event* tersebut melalui *subscription* yang dapat diakses kapan saja [2]. Sebagai contoh, *subscriber* dapat menyatakan ketertarikannya pada semua *event* terkait tentang sebuah buku, seperti ketersediaan dari edisi baru atau *update* edisi baru buku tersebut pada situs web terkait.

Sistem *publish/subscribe* memiliki arsitektur yang terdiri dari 3 lapisan, yaitu lapisan *network protocol*, *overlay network*, dan *event routing*. Lapisan *overlay network* merupakan lapisan yang paling penting karena terdiri dari *broker network*.



Gambar 1. Komunikasi *Point-to-Point*

Hal ini terjadi karena *broker network* merupakan penghubung langsung antara *publisher* dan *subscriber*, seperti terlihat pada Gambar 2 [1].

E. Protokol MQTT (*Message Queuing Telemetry Transport*)

Pada tahun 1999, Andy Stanford-Clark dari IBM berkolaborasi dengan Alen Nipper dari Eurotech membuat spesifikasi protokol pesan yang baru yaitu MQTT. MQTT adalah protokol konektivitas *machine-to-machine (M2M)/“Internet of Things.”* MQTT yang sering digunakan sebagai broker pada sistem *publish/subscribe* yang bersifat sangat ringan dan mudah untuk diimplementasikan [3]. Karakteristik ini membuat MQTT ideal digunakan dalam lingkungan jaringan yang terbatas, seperti: *bandwidth* yang rendah dan tidak terjamin dan prosesor atau memori yang terbatas pada suatu sistem *embedded*.

Protokol MQTT memiliki 14 jenis tipe pesan yang berbeda-beda, seperti terlihat pada Tabel 1 [5].

Protokol MQTT yang digunakan dalam penelitian ini adalah protokol MQTT v3.1. Cara kerja dari protokol MQTT v3.1 adalah sebagai berikut:

1. Connection Setup

Connection setup adalah bagian awal yang dilakukan *publisher*, broker, dan *subscriber* untuk membangun komunikasi.

2. Disconnecting

Disconnecting terjadi saat *subscriber* berhenti berlangganan pada suatu topik yang dipublikasikan oleh *server*.

3. Heart Beat

Heart beat adalah tahap antara *publisher*, broker, dan *subscriber* sebelum *disconnect*.

4. Subscribing

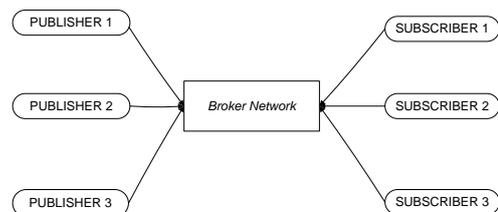
Tahap ini terjadi pada saat *subscriber* akan berlangganan satu atau beberapa topik dari *server*.

5. Publishing Message

Server akan mengirimkan topik sesuai dengan permintaan klien. *Server* mempublikasikan topik kepada klien menggunakan *level QoS*, seperti terlihat pada Gambar 3, Gambar 4, dan Gambar 5.

F. Quality of Service (QoS)

Quality of Service adalah kemampuan jaringan untuk menyediakan layanan yang lebih baik lagi bagi trafik yang melewatinya [2].



Gambar 2. Sistem *publish/subscribe*

QoS terdiri dari *delay*, *jitter*, dan *bandwidth*. *Delay* adalah waktu tunda suatu paket yang diakibatkan oleh proses transmisi dari suatu titik ke titik lain yang menjadi tujuan, *jitter* merupakan variasi *delay* yang terjadi pada jaringan IP, dan *bandwidth* adalah lebar frekuensi yang digunakan oleh sinyal dalam mengirimkan paket data.

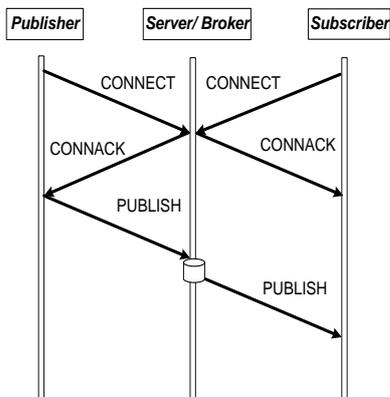
Protokol MQTT memiliki 3 level QoS, yaitu:

1. *QoS level 0 : at most once delivery*

Pesan dikirim dengan menggunakan jaringan TCP/IP. Respon atau jawaban dari klien tidak diminta dan tidak akan ada pengiriman ulang pesan. Pesan akan diterima hanya sekali atau tidak sama sekali.

TABEL 1
MESSAGE TYPE

Mnemonic	No.	Description
CONNECT	1	Klien request to connect to Server
CONNACK	2	Connect Acknowledgment
PUBLISH	3	Publish message
PUBACK	4	Publish Acknowledgment
PUBREC	5	Publish Received-assured delivery part 1
PUBREL	6	Publish Release-assured delivery part 2
PUBCOMP	7	Publish Complete-assured delivery part 3
SUBSCRIBE	8	Klien Subscribe request
SUBACK	9	Subscribe Acknowledgment
UNSUBSCRIBE	10	Klien Unsubscribe request
UNSUBACK	11	Unsubscribe Acknowledgment
PINGREQ	12	PING Request
PINGRESP	13	PING Response
DISCONNECT	14	Klien is Disconnecting



Gambar 3. Publishing message with Qos level 0

2. *QoS level 1 : at least once delivery*

Pesan akan dikirim oleh *publisher*. Jika terjadi kegagalan pengiriman pesan atau pesan konfirmasi tidak diterima, *publisher* akan mengirim pesan dengan menyertakan bit DUP di bagian kepala pesan.

3. *QoS level 2 : exactly once delivery*

QoS level 1 menjamin bahwa pesan duplikat yang dikirim akan diterima oleh *server*. QoS level 2 digunakan ketika duplikat pesan yang dikirim tidak diterima oleh *server*.

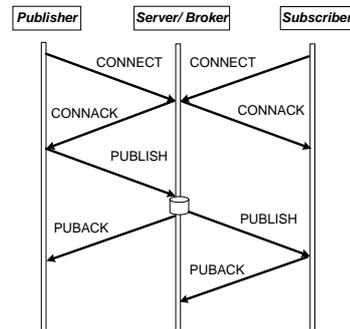
G. Perbedaan Antara Protokol MQTT dan DDS

Protokol MQTT dan DDS memiliki beberapa kesamaan fungsi. Beberapa diantaranya adalah menggunakan sistem *publish/subscribe*, bersifat *real time* dan *machine to machine* sehingga digunakan dalam pengembangan *Internet of Things*, dan mendukung penggunaan pada *mobile phone* dan sistem *embedded*. Dari beberapa kesamaan fungsi tersebut, terdapat beberapa perbedaan utama antara kedua protokol ini, seperti terlihat pada Tabel 2 [4].

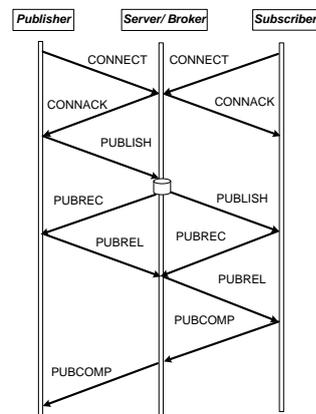
III. ANALISA DAN PERANCANGAN SISTEM

A. Langkah-langkah Percobaan Sistem

Penelitian ini menggunakan 8 langkah, seperti terlihat pada Gambar 6. Tujuan pembuatan langkah-langkah penelitian ini adalah agar dalam proses pengerjaan penelitian ini tetap berada pada jalurnya, sehingga tujuan penelitian ini dapat dicapai.



Gambar 4. Publishing message with Qos level 1



Gambar 5. Publishing message with Qos level 2

B. Emulasi Jaringan

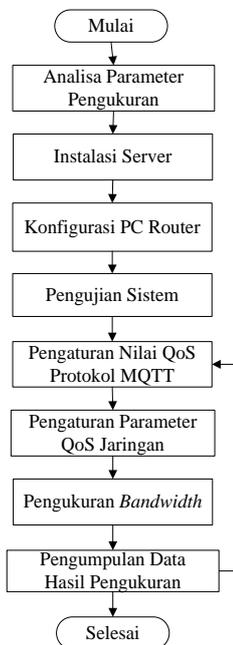
Perancangan emulasi jaringan untuk penelitian ini dapat dilihat pada Gambar 7.

C. Perancangan Arsitektur Publish/Subscribe System

Perancangan sistem pada penelitian ini membutuhkan 2 buah PC yang memiliki 2 buah *interface* LAN, seperti terlihat pada Gambar 8. PC tersebut akan bertindak sebagai *broker* dan PC *router*. Selain itu, dibutuhkan juga 2 buah PC yang akan bertindak sebagai klien, yaitu sebagai *publisher* dan *subscriber*.

TABEL 2
PERBEDAAN PROTOKOL MQTT DENGAN DDS

Protokol Perbedaan	MQTT	DDS
<i>Messaging Uses</i>	<i>Telemetry</i> : perangkat ke server, data terpusat, <i>IT cloud</i> Terpusat pada server	<i>Intelligent System: real time cloud</i> . Sistem terdistribusi terintegrasi dan tertanam
<i>Deployment</i>	<i>Point of sale (POS)</i> untuk kasino dan pusat perbelanjaan, dan <i>mobile device (Facebook essanger)</i>	Mengontrol robot dan kendaraan tanpa kemudi
<i>Architecture</i>	Terpusat: semua pesan dialirkan ke broker dan TCP protokol,	<i>Desentralisasi</i> dan UDP protokol,
<i>Quality of Service (QoS)</i>	Berfokus pada pengiriman pesan, yaitu <i>at most once, at least once, dan exactly once</i>	Berfokus pada toleransi kesalahan, waktu, dan tidak memiliki ketergantungan



Gambar 6. *Flowchart* Percobaan Penelitian

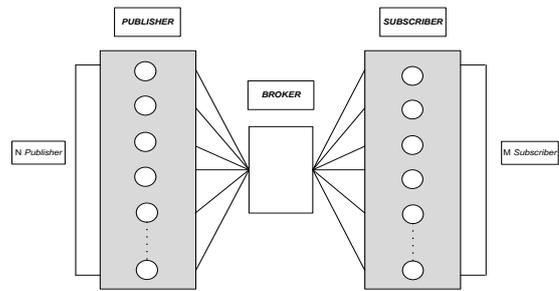
IV. IMPLEMENTASI DAN HASIL PENGUKURAN

Data hasil pengukuran yang telah dilakukan akan diolah menjadi grafik, sehingga penyajiannya lebih mudah untuk dipahami. Grafik yang dihasilkan akan dianalisis untuk setiap QoS, sehingga dapat ditarik suatu kesimpulan yang mengarah pada pengaruh QoS protokol MQTT pada QoS jaringan, yaitu *bandwidth, delay, jitter, dan packet loss*.

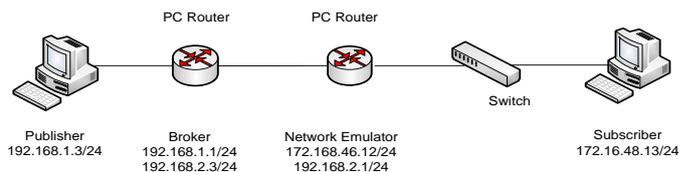
A. Metode Pengukuran Parameter QoS Sistem Publish/Subscribe

Parameter QoS dari sistem *publish/subscribe* yang akan diukur pada penelitian ini, meliputi *bandwidth, delay, jitter, dan packet loss*. Hasil dari pengukuran tersebut akan disajikan dalam bentuk tabel dan akan diolah menjadi grafik. Data yang diperoleh berasal dari *Collectd* (salah satu *daemon* pada linux yang dapat mengumpulkan statistik performa sistem secara periodik) yang dipasangkan pada broker sehingga grafik yang dihasilkan adalah berdasarkan kedua antar muka yang terpasang pada broker, yaitu *ethernet 0* (terhubung ke *subscriber*) dan *ethernet 1* (terhubung ke *publisher*).

1. Melakukan pengatur untuk parameter *bandwidth, delay, jitter, dan packet loss* pada NetEm, seperti `tc qdisc add dev eth0 root netem loss 0%`, `tc qdisc change dev eth0 root netem loss 1%`, `tc qdisc add dev eth1 root netem delay 0ms`, dan `tc qdisc change dev eth0 root NetEm delay 80ms`
2. Menjalankan `localhost/collectd` pada broker seperti terlihat pada Gambar 9.
3. Mengeksekusi *script bash* yang telah dibuat untuk dapat menjalankan *publisher* dan *subscriber* dalam jumlah banyak.
4. Perintah yang digunakan untuk mengeksekusi coding *script bash* adalah `./langganan.sh` dan `./terbit.sh`. Gambar 10 menunjukkan pada saat `./langganan.sh` dieksekusi.

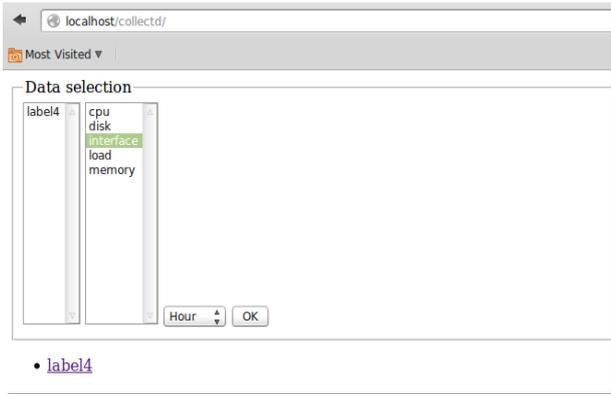


Gambar 7. Emulasi Jaringan

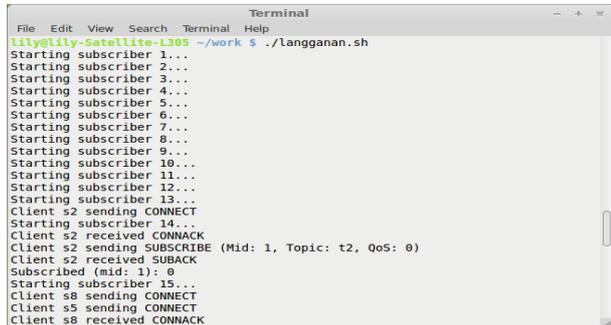


Gambar 8. Perancangan Arsitektur *Publish/Subscribe* System

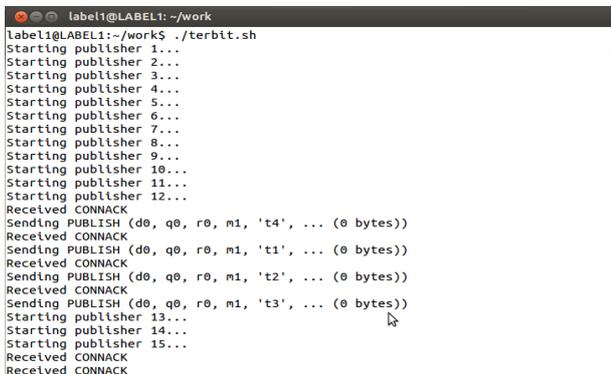
5. Pada saat `./langganan.sh` dieksekusi maka `mosquitto_sub` akan dijalankan dan akan menunggu sampai `publisher` mempublikasikan topik yang dipesan. Sementara itu, Gambar 11 ini adalah pada saat `./terbit.sh` di eksekusi.
6. Menyimpan hasil *capture* data oleh `Collectd` seperti terlihat pada Gambar 12. Data yang akan dimasukkan dalam *MS. Excel* adalah 3,3k yang merupakan rata-rata paket keluar yang dilalui pada *ethernet* 0.
7. Menghapus directory `Collectd` pada `var/lib/collectd/rrd/label4/interface-eth0` dan interface `-eth1`.
8. Hasil *capture* data yang telah diperoleh dimasukkan dalam *ms excel*.



Gambar 9. Tampilan localhost/collectd



Gambar 10. Eksekusi *scripts bash* `./langganan.sh`



Gambar 11. Eksekusi *scripts bash* `./terbit.sh`

B. Perbandingan Hasil Pengukuran Konsumsi Bandwidth sebagai Pengaruh Packet Loss untuk QoS Protokol MQTT

Hasil pengukuran konsumsi *bandwidth* sebagai pengaruh *packet loss* 25% untuk QoS 0, 1, dan 2 dimunculkan dalam satu grafik dapat dilihat pada Gambar 13.

Hasil pengukuran konsumsi *bandwidth* tertinggi sebagai pengaruh *packet loss* 25% untuk QoS 0, 1, dan 2 terjadi pada saat QoS 2, yaitu 112,3 *kilo bits/second* yang disusul dengan *packet loss* dari QoS 1, yaitu 87,6 *kilo bits/second* dan *packet loss* dari QoS 0, yaitu 81,9 *kilo bits/second* dengan nilai rasio 1,38.

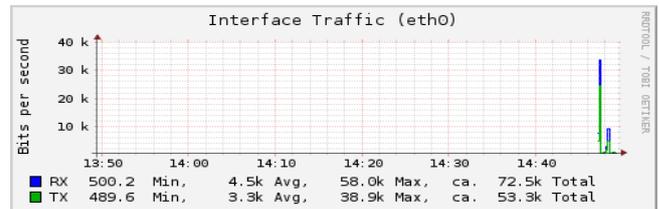
C. Perbandingan Hasil Pengukuran Konsumsi Bandwidth sebagai Pengaruh Delay untuk QoS Protokol MQTT

Hasil pengukuran konsumsi *bandwidth* sebagai pengaruh *delay* 500 ms untuk QoS 0, 1, dan 2 dimunculkan dalam satu grafik dapat dilihat pada Gambar 14.

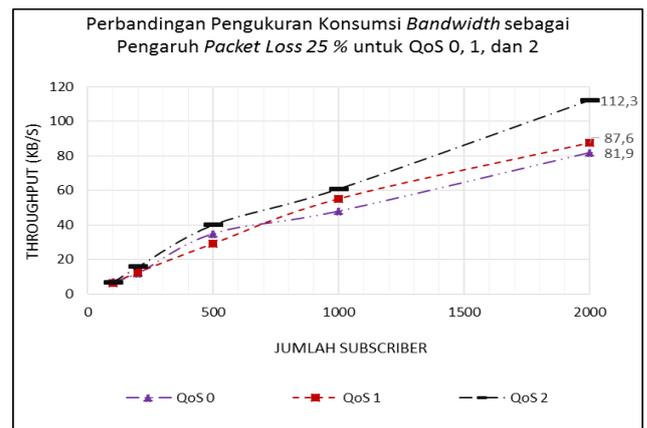
Hasil pengukuran konsumsi *bandwidth* tertinggi sebagai pengaruh *delay* 500 ms untuk QoS 0, 1, dan 2 terjadi pada saat QoS 2, yaitu 112,5 *kilo bits/second* yang disusul dengan *delay* dari QoS 1, yaitu 91,4 *kilo bits/second*, dan *delay* dari QoS 0, yaitu 80,2 *kilo bits/second*.

D. Perbandingan Hasil Pengukuran Konsumsi Bandwidth sebagai Pengaruh Jitter untuk QoS Protokol MQTT

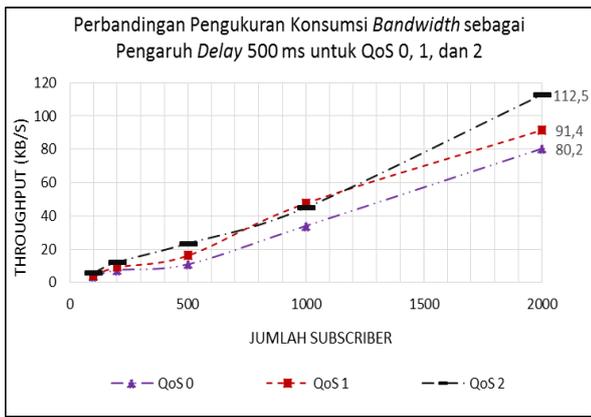
Hasil pengukuran konsumsi *bandwidth* sebagai pengaruh *jitter* 70 ms untuk QoS 0, 1, dan 2 dimunculkan dalam satu grafik dapat dilihat pada Gambar 15.



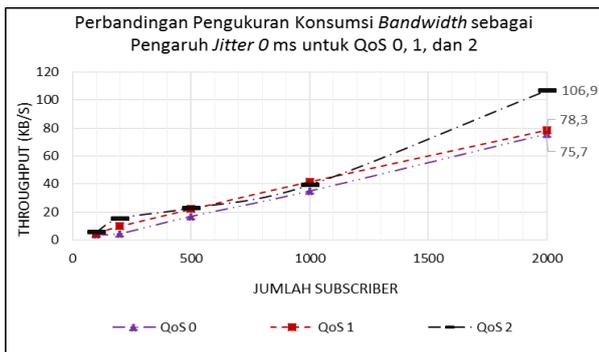
Gambar 12. Screen Shoot Collected



Gambar 13. Grafik Hasil Perbandingan Pengukuran Konsumsi *Bandwidth* sebagai Pengaruh *Packet loss* untuk QoS 0, 1, dan 2



Gambar 14. Grafik Hasil Perbandingan Pengukuran Konsumsi *Bandwidth* sebagai Pengaruh *Delay* untuk QoS 0, 1, dan 2



Gambar 15. Grafik Hasil Perbandingan Pengukuran Konsumsi *Bandwidth* sebagai Pengaruh *Jitter* untuk QoS 0, 1, dan 2

Hasil pengukuran konsumsi *bandwidth* tertinggi sebagai pengaruh *jitter* untuk QoS 0, 1, dan 2 terjadi pada saat QoS 2, yaitu 106,9 kilo *bits/second* yang disusul dengan *jitter* dari QoS 1, yaitu 78,3 kilo *bits/second*, dan *jitter* dari QoS 0, yaitu 75,7 kilo *bits/second*.

V. KESIMPULAN

Dari hasil uji implementasi yang telah dilakukan maka dapat diambil kesimpulan bahwa:

1. QoS protokol MQTT tidak berpengaruh pada QoS jaringan, yaitu *bandwidth*, *delay*, *jitter*, dan *packet loss* pada jaringan *publisher* (*Ethernet 0* pada broker) karena tidak dilakukan pengaturan nilai untuk *bandwidth*, *delay*, *jitter*, dan *packet loss* pada *Ethernet 0* melalui *NetEm*. Sementara itu, QoS protokol MQTT sangat berpengaruh pada QoS jaringan, yaitu *bandwidth*, *delay*, *jitter*, dan *packet loss* pada jaringan *subscriber* (*Ethernet 1* pada broker) karena dilakukan pengaturan nilai untuk *bandwidth*, *delay*, *jitter*, dan *packet loss* pada *Ethernet 1* melalui *NetEm*.
2. Semakin tinggi *level* QoS protokol MQTT (pengiriman pesan akan semakin terjamin) maka konsumsi *bandwidth* sebagai pengaruh QoS jaringan juga meningkat, yaitu QoS 2 memiliki konsumsi *bandwidth* tertinggi yang disusul dengan QoS 1 dan QoS 0.
3. Implementasikan QoS *level* 2 untuk komunikasi yang tidak mentoleransi terjadinya kegagalan dan duplikasi

pengiriman pesan. Implementasikan QoS *level* 1 untuk komunikasi yang dapat mentoleransi terjadinya duplikasi pengiriman pesan dan implementasikan QoS *level* 0 jika kapasitas *bandwidth* yang tersedia untuk melakukan komunikasi antar setiap *subscriber* sangat terbatas atau minim.

REFERENSI

- [1] A. I Made. "Alasan Sistem Terdistribusi." Sistem Terdistribusi, Bandung, pp. 2-3, 2010.
- [2] A. Kevin. "RFID Journal." *That 'Internet of Things' Thing*, pp.4-5, 2011, Nov. 2011 .
- [3] B. Brad *et al.*, "The Internet of Things." *The Interconnecting of Everything*, pp. 1, 2013.
- [4] B. David. "MQTT and DDS Comparison." Internet: <http://www.slideshare.net/RealTimeInnovations/comparison-of-mqtt-and-dds-as-m2m-protocols-for-the-internet-of-things>, May. 2013 [Okt.19, 2013].
- [5] C. George. "Indirect Communication," in *Distributed System Concepts and Design*, fifth edition, 2012, pp. 229-253.
- [6] Fatoni. "Metodologi Penelitian." Analisis Kualitas Layanan Jaringan Intranet (Studi Kasus Universitas Bina Darma), Palembang, pp. 3-6.
- [7] IBM. "MQTT V3.1 Protocol Specification." Internet: http://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/MQTT_V3.1_Protocol_Specific.pdf, [Nov.5, 2013].
- [8] IBM. "Quick Tour." Internet: https://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.wmb_v6/wmb/6.0/Overview.html, [Okt.24, 2013].
- [9] IBM. "The Internet of Things Helps Enable Proactive Data Access From Any Connected Device." Internet: <http://www-01.ibm.com/software/info/internet-of-things/>, [Okt.21, 2013].
- [10] K. Ariane. "Brief Introduction." *Manual Tc Packet Filtering and NetEm*, pp. 3-4, Jul. 20, 2006.
- [11] K. Dustin. "MQTT-MQ Telemetry Transport." Internet: <http://manpages.ubuntu.com/manpages/raring/man7/mqtt.7.html#content0>, Feb. 05, 2012 [Nov. 05, 2013].
- [12] L. Valerie *et al.*, "Building a Smarter Planet," in *Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry*, first edition, September 2012, pp. 2-3.
- [13] Membangun Aplikasi Java EE Menggunakan Messaging, 10th ed., Tanmia Informatika, Jakarta Selatan, pp. 5-6.
- [14] OMG. "Data Distribution Service (DDS)." Internet: <http://www.omg.org/hot-topics/dds.htm>, Okt. 12, 2013 [Nov.15, 2013].
- [15] RTI. "RTI Data Distribution Service." Internet: http://www.linuxworks.com/partners/show_product.php?ID=88, [Okt.22, 2013].
- [16] S. Vatsal. "Mosquitto." Internet: <http://mqtt.org/wiki/doku.php/brokers#mosquitto>, Okt. 11, 2013 [Okt.10, 2013].
- [17] Unknown. "MQ Telemetry Transport." Internet: <http://mqtt.org/>, [Okt.22, 2013].

Salsalina O. F. Tarigan, lulusan Sistem Komputer ITHB tahun 2014. Minat penelitian: *computer networking*.

Herry Imanta Sitepu, menempuh pendidikan S1 di Teknik Elektro ITB dan lulus tahun 1999, dan memperoleh gelar magister dan doktor di jurusan yang sama di ITB. Sejak tahun 2006 aktif sebagai pengajar di Prodi Sistem Komputer ITHB. Minat penelitian: *computer networking*, *programming* dan *distributed system*.

Maclaurin Hutagalung, menerima gelar Sarjana Teknik Elektro dari ITB, gelar Magister Sains dari University of Twente Belanda di bidang Sinyal, Sistem, dan Kendali, dan gelar Doktor dari Tokyo Institute of Technology di bidang Sistem Kendali Nonlinear. Sejak tahun 2012 aktif sebagai pengajar di Prodi Sistem Komputer ITHB. Minat penelitian pada Kendali Sistem Dinamis, Robotika, dan Penerbangan.